qac

October 7, 2025

[76]:
```
import pandas as pd
import numpy as np
import quaternion

import matplotlib.pyplot as plt

filename = 'data/IMU_Data_4.csv'

df = pd.read_csv(filename)
df['Gyro_w'] = 0
df[['Gyro_x', 'Gyro_y', 'Gyro_z']] *= np.pi / 180

euler_arr = df[['Euler_x', 'Euler_y', 'Euler_z']].to_numpy() * np.pi / 180
omega_q_arr= quaternion.as_quat_array(df[['Gyro_w', 'Gyro_x', 'Gyro_y',␣
 ↪'Gyro_z']])
q_truth_arr = quaternion.as_quat_array(df[['Quat_0', 'Quat_1', 'Quat_2',␣
 ↪'Quat_3']])

f = 100 #Hz
dt = 1/f

time = np.arange(len(df)) / f
view_window = slice(0, 3000)

truth_components = q_truth_arr.view(np.float64).reshape(-1, 4)
```

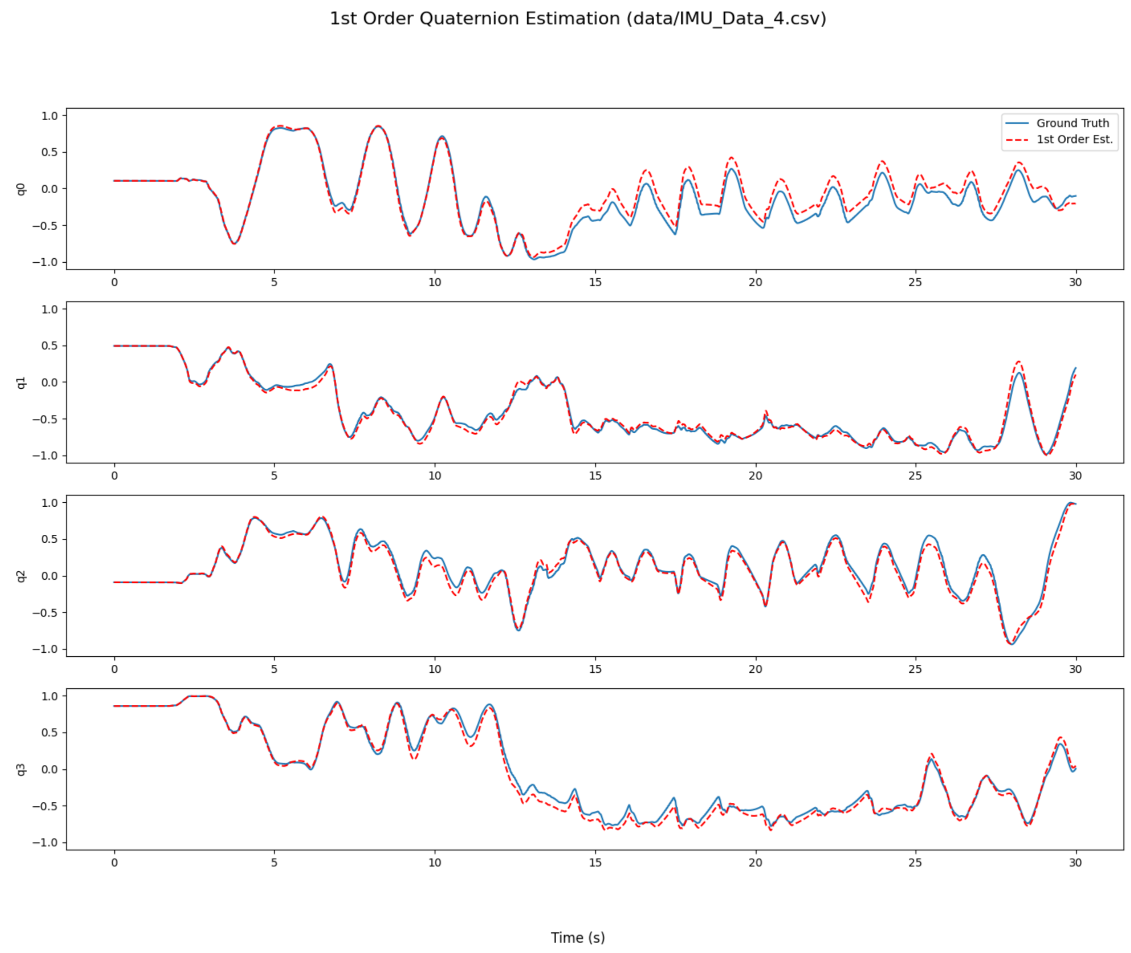
1 Scheme 1: First Order Approx.

Taylor Series:

$$\begin{aligned} q_{t+1} &= q_t + \dot{q}_t \Delta t + O(\Delta t) \\ &\approx q_t + \frac{\Delta t}{2} q_t \otimes \omega_q \end{aligned} \tag{1}$$

[77]:
```
f = 100 #Hz
dt = 1/f

q_prev = q_truth_arr[0]
```

```
q_est_1st_arr = [q_prev]

for omega_q in omega_q_arr[1:]:
    q_dot =  0.5  * q_prev * omega_q
    q = q_prev + q_dot * dt
    q = q.normalized()

    q_est_1st_arr.append(q)
    q_prev = q

q_est_1st_arr = np.array(q_est_1st_arr)

est_1st = q_est_1st_arr.view(np.float64).reshape(-1, 4)

fig, ax = plt.subplots(4, 1, figsize=(17,12))

for i in range(4):
    ax[i].plot(time[view_window], truth_components[view_window, i], label='Ground Truth ')
    ax[i].plot(time[view_window], est_1st[view_window, i], label='1st Order Est.', linestyle='dashed', color='red')
    ax[i].set_ylabel(f'q{i}')
    ax[i].set_ylim([-1.1, 1.1])

ax[0].legend(loc='upper right')
fig.suptitle(f'1st Order Quaternion Estimation ({filename})', fontsize=16)
fig.supxlabel('Time (s)');
```



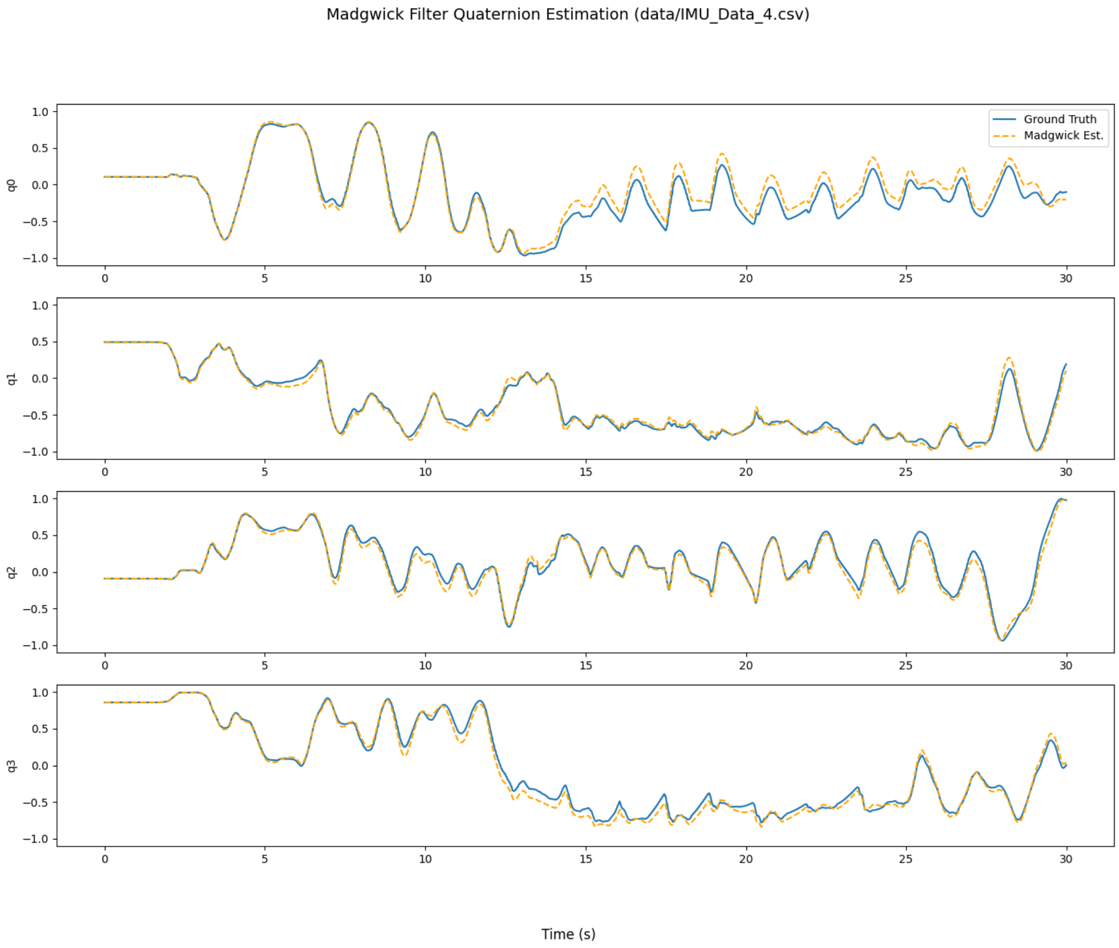
2 Scheme 2: Madgwick Filter

- We should avoid this, because Madgwick does not perform well under high acclerations.
- For demonstration purposes.

[78]:

```
from ahrs.filters import Madgwick
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np

omega = df[['Gyro_x', 'Gyro_y', 'Gyro_z']].to_numpy() * np.pi / 180
acc = df[['Acc_x', 'Acc_y', 'Acc_z']].to_numpy()
mag = df[['Mag_x', 'Mag_y', 'Mag_z']].to_numpy()

madgwick = Madgwick(gyr=omega, acc=acc, mag=mag, frequency=f, beta=0.5)

q_est_madgwick = -madgwick.Q
```

```
fig, ax = plt.subplots(4, 1, figsize=(17,12))


for i in range(4):
    ax[i].plot(time[view_window], truth_components[view_window, i],␣
 ↪label='Ground Truth ')
    ax[i].plot(time[view_window], est_1st[view_window, i], label='Madgwick Est.
 ↪', linestyle='dashed', color='orange')
    ax[i].set_ylabel(f'q{i}')
    ax[i].set_ylim([-1.1, 1.1])

ax[0].legend(loc='upper right')
fig.suptitle(f'Madgwick Filter Quaternion Estimation ({filename})', fontsize=14)
fig.supxlabel('Time (s)');
```


Madgwick Filter Quaternion Estimation (data/IMU_Data_4.csv)
Ground Truth
Madgwick Est.
q0
q1
q2
q3
Time (s)